BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B70 01005

SUBJECT: Onboard Computing Capability Required to Compress AAP Telemetry Data - Case 620/900

DATE: January 7, 1970

FROM: D. O. Baechler
R. J. Pauly

ABSTRACT

This memorandum assesses the feasibility of compressing the AAP real-time telemetry bit streams using an onboard computer that is currently available at MSC. The two 51.2 KBPS and one 72 KBPS real-time bit streams planned for the AAP could be compressed from their total of 174.4 KBPS to about 20 to 35 KBPS. The computer requirement to accomplish this is estimated to be 19,000 memory words and 266,000 operations per second. An IBM 4 Pi/EP would not be fast enough to do this job, but it would be able to compress and merge the two 51.2 KBPS streams. It would, however, require the CPU to operate within 1% of its total speed capability.

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B70 01005

SUBJECT: Onboard Computing Capability Required to Compress AAP Telemetry Data - Case 620/900

DATE: January 7, 1970

FROM: D. O. Baechler
R. J. Pauly

MEMORANDUM FOR FILE

Introduction

The objective of this memorandum is to estimate the onboard computing capability that would be required to compress and combine two 51.2 KBPS and one 72 KBPS AAP bit streams and to determine the adequacy of an available computer to do this task.*

The GSFC Network Computing Branch has recently studied the feasibility of compressing AAP telemetry data using the UNIVAC 642B digital computer.[1] This study has been used in preparing the estimates in this paper. Additional data compression studies by APL[2] and Mitre[3] have been used to support these estimates.

An average compression ratio for AAP data of 20-1 appears feasible based on the studies which have been performed.[1,2,3] Thus, the average information output rate for the three compressed bit streams would be 8.72 KBPS. This would be increased by a factor of perhaps three or four to allow for identification, error coding, and other overhead data. The resulting average data output rate would be about 20 to 35 KBPS.

Computer Power

The GSFC study concludes that using a zero order compression algorithm the UNIVAC 642B could process AAP data at an input rate of 15,000 samples per second with 80% CPU loading.

With this information, the onboard computer power required to compress and combine two 51.2 KBPS and one 72 KBPS AAP bit streams can be estimated as follows:

---

* An IBM 4 Pi/EP which is available at MSC has been suggested as a candidate for the task.

1. The typical instructions executed by the U642B in processing the AAP data would be add, subtract, store, shift and compare; each of these has an execution time of 4μ seconds, giving an execution rate of 250,000 instructions per second.

2. Therefore, the number of instructions executed per sample would be,

$$\frac{80\% \times 250{,}000 \text{ instructions per second}}{15{,}000 \text{ samples per second}} = 13.3 \text{ instructions per sample}$$

The reasonableness of this number is supported by other studies. APL[2] estimated, based on preliminary programming of a somewhat similar task that about 30% loading of the U642B would be required to compress 6400 samples per second. This is

$$\frac{30\% \times 250{,}000 \text{ instructions/second}}{6400 \text{ samples/second}} \approx 12 \text{ instructions per sample}$$

The task considered by APL uses a fixed window in its preliminary form and it compresses a single stream of data.

A Bellcomm study[4] concludes that a zero-order compression algorithm would require 10 operations per sample, as a minimum. The block diagram used to arrive at this estimate is shown in Figure 1.

Therefore, 13.3 instructions per sample will be used in subsequent calculations.

3. The samples per second to be processed would be,

$$\frac{51.2 \times 10^3 \text{ bits per second}}{8 \text{ bits per sample}} = 6400 \text{ samples per second for each 51.2 KBPS link}$$

$$\frac{72 \times 10^3 \text{ bits per second}}{10 \text{ bits per sample}} = 7200 \text{ samples per second for the 72 KBPS link}$$

Total = 6400 + 6400 + 7200 = 20,000 samples per second

4. Thus the required onboard computer instruction rate would be,

13.3 instructions per sample x 20,000 samples per second = 266,000 instructions per second.

5. Allowing a 20% CPU margin, the onboard computer's typical instruction execution time would be,

$$\frac{266,000}{80\%} = 330,000 \text{ instructions per second, or } 3\mu \text{ sec/instruction.}$$

Computer Storage

The U642B core storage requirements developed in the GSFC study were,

Function	Words
Input Processor	3,000
Output Processor	500
Compressor	1,500
Operating System	9,000
Buffers-input, output, queues	32,000

The allocation of 32,000 words for buffer storage was based on making the maximum practical use of the U642B core storage that was remaining after the program modules were sized. The minimum buffer storage that would be needed can be estimated as follows:

1. Two input buffers would be needed for each bit stream, each buffer would hold a frame of data. The data frame in one buffer would be processed at the same time that another frame was being entered into the other buffer. Thus the input buffer storage would be,

   51.2 KBPS link = 2 buffers per frame x 128 words per frame = 356 words

   72 KBPS link = 2 buffer per frame x 60 words per frame = 120 words

   Two 51.2 KBPS links + one 72 KBPS link = 356 + 356 + 120 = 832 words

2. Assuming as the worst case that the output would be the same as the input (i.e., no compression) the output buffer storage would be 832 words.

3. Allowing internal queue storage to be the same as the input buffer storage, the queue storage would be 832 words.

4. Allowing internal storage of the last value of each of the approximately 2,000 parameters that would be tested, the requirement would be 2,000 words.

5. The resulting total buffer storage requirements would be 4,496 words.

An estimate of the total onboard computer core storage requirements can be obtained by combining the U642B program module sizes with buffer sizes developed in this report. The result is a total core storage requirement of approximately 19,000 words. The U642B has a 30 bit word length. However, a computer word length of 18 to 24 bits would be adequate to perform the required operations on the 8 bit and 10 bit AAP telemetry words.

Adequacy of an Available Aerospace Computer

It would be necessary for an onboard computer to have 19,000 words of memory and to perform 330,000 operations per second to perform the task described above. The IBM 4 Pi/EP has been suggested for this application. It is obtainable with 16K, 24K, or 32K of memory, as well as other 8K increments from 8K to 64K on a single bus. It performs a fixed-point register to memory add operation in 5.8μ sec; this is about 172,000 operations per second assuming the add operation to be a typical operation. This speed is not adequate to perform the task described above.

Two questions immediately arise: (1) is this a realistic analysis, and (2) is there any possibility that the IBM 4 Pi/EP could perform this task or some part of it?

(1) It is a simplification, although usually a realistic one, to assume that an add operation is a typical operation. Special input/output features, the speed of other operations, differences in instruction repertoire, and word length are among the features not considered when this assumption is made. For instance, the IBM 4 Pi/EP is capable of performing register-to-register add or compare operations in 2.3μ seconds and register-to-register logic operations in about 1.5μ seconds. The U642B has no register-to-register instructions in its repertoire. This may be an important difference, depending on the task being done. Because of the nature of the merging and compression task being considered here, there will probably be few register-to-register operations. Of the 12 to 14 instructions performed

for each sample, there might be 3 or 4 register-to-register operations that result from indexing needs. A realistic assumption is that one in four instructions are register-to-register operations. Then the average instruction execution time is [(1x2.3) + (3x5.8)]/4 = 4.9μ seconds or about 200,000 operations per second. A more optimistic assumption is that one out of every three instructions are register-to-register operations. For this case, the average instruction execution time is [(1x2.3) + (2x5.8)]/3 = 4.63μ seconds or 216,000 operations per second.

The table below summarizes these results:

	PERFORMANCE (OPERATIONS/SECOND)	
Criteria	IBM 4 Pi/EP	U642B
Add Times Only	172,000	250,000
1/4 Instruction Mix	200,000	250,000
1/3 Instruction Mix	216,000	250,000

(2) Consider the implications of each of these results. If add times are used as the criteria for comparing the two computers, the IBM 4 Pi/EP is capable of processing

$$\frac{172{,}000 \text{ ops/sec}}{13.3 \text{ ops/sample}} = 13{,}000 \text{ samples/sec}$$

with 100% CPU loading. The least demanding job is to merge the two 51.2 KBPS data streams, which would require processing of 6400 x 2 = 12,800 samples per second. This results in 99% CPU loading.

If a 1 out of 4 mix of register-to-register and register-memory instructions is assumed, the EP can process 200,000/13.3 = 15,000 samples per second with 100% CPU loading. If it merges the two 51.2 KBPS streams, the result is 86% CPU loading. If the 72 KBPS and one of the 51.2 KBPS streams are merged, the result is 13,800/14,900 = 93% CPU loading.

If a 1 out of 3 mix of register-to-register and register-memory instructions is assumed, 216,000/13.3 = 16,200 samples per second can be processed by the EP with 100% CPU loading. But even this is not sufficient to process all three streams.

Conclusions

It may be practical to merge any two of the three data streams, but the IBM 4Pi/EP would have to be entirely dedicated to this task. It is extremely time critical, and may result in 99% CPU loading even for merging the two 51.2 KBPS streams. It may be possible to merge the 72 KBPS stream with one of the 51.2 KBPS streams, but it does not appear feasible to merge all three streams.

D. O. Baechler

R. J. Pauly

1032-$\frac{DOB}{RJP}$-dmu

Attachment

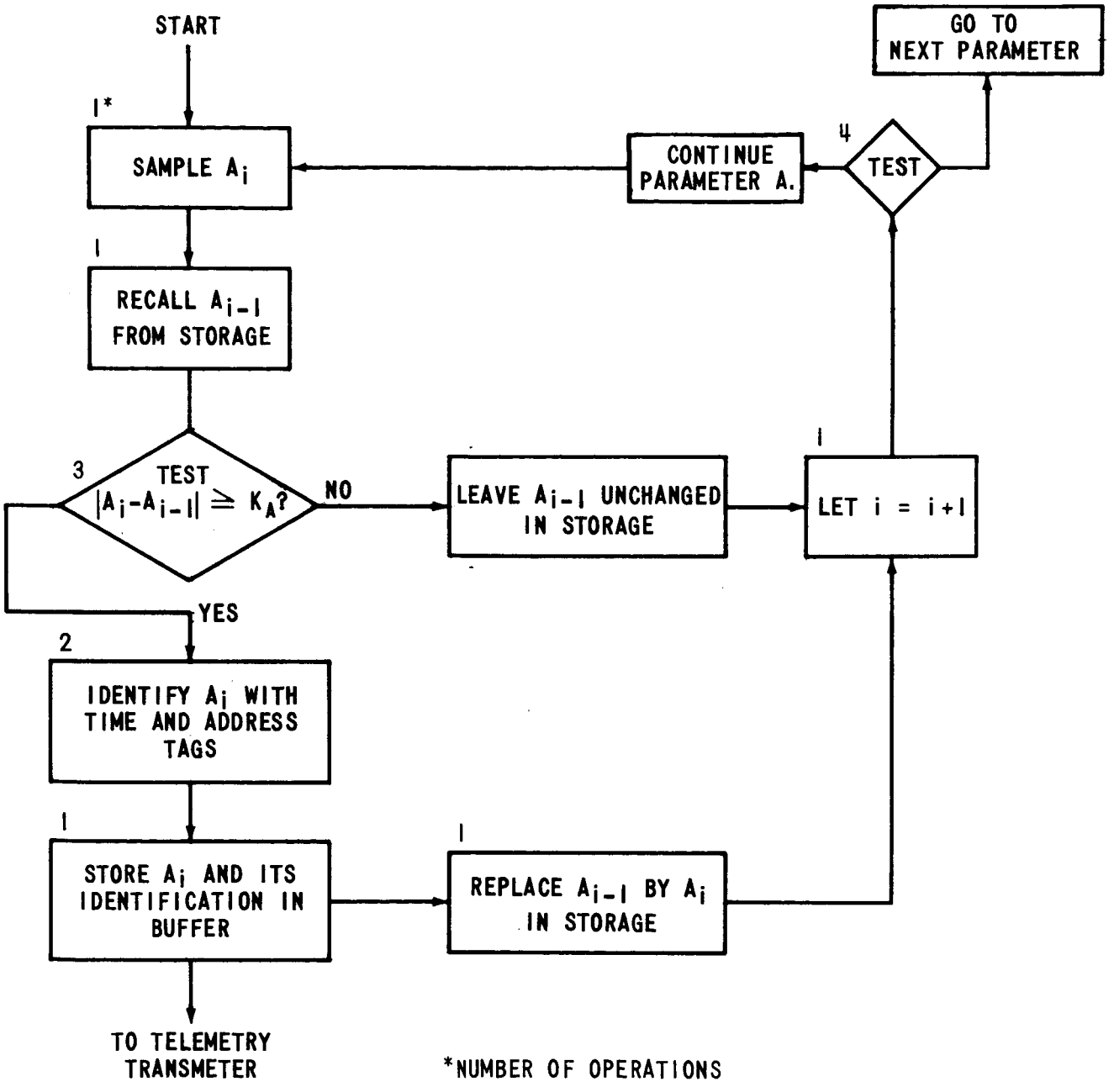

FIGURE 1 - BLOCK DIAGRAM OF STEPS IN DATA COMPRESSION USING ZERO-ORDER PREDICTOR TECHNIQUE

REFERENCES

1. "Data Redundancy Removal at Remote Sites of the Manned Space Flight Network," Network Computing Branch, Goddard Space Flight Center, Technical Report, September 1969.

2. "A Real-Time Data Compression System for Apollo PCM Telemetry," Space Communications Group, Applied Physics Laboratory, Special Report CSC-1-150, September 1969.

3. Barrow, J.T., L. L. Stine, "The Application of Data Compaction to Apollo Flight Control Telemetry Data," Mitre Technical Report, BTR-498, October 1, 1967.

4. Baechler, D. O., "The Potential for Computer Use in Spaceborne Experiments," Bellcomm Memorandum for File, March 11, 1969.

BELLCOMM, INC.

SUBJECT: Onboard Computing Capability Required to Compress AAP Telemetry Data

FROM: D. O. Baechler
R. J. Pauly

Distribution List

NASA Headquarters

L. X. Abernethy - MOP
H. Cohen - MLR
P. E. Culbertson - MLA
J. H. Disher - MLD
W. B. Evans - MLO
L. K. Fero - MLV
J. P. Field, Jr. - MLP
T. A. Keegan - MA-2
A. W. Kinny - MOP
M. Savage - MLT
W. C. Schneider - ML
J. D. Stevenson - MO

MSC

V. P. Neshyba - KA

GSFC

W. S. Guion - 813
P. J. Pashby - 813

Bellcomm, Inc.

G. M. Anderson
A. P. Boysen, Jr.
D. A. DeGraaf
J. H. Fox
D. R. Hagner
J. J. Hibbert
B. T. Howard
J. E. Johnson
B. W. Kim
W. J. Martin
J. Z. Menard
J. M. Nervik
I. M. Ross
R. L. Selden
W. Strack
J. W. Timko
R. L. Wagner
A. G. Weygand
M. P. Wilson

Central Files
Department 1024 Files
Library